



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/976,313	10/11/2001	Williams L. Harrison III	42390P12907	1914

8791 7590 11/02/2005

BLAKELY SOKOLOFF TAYLOR & ZAFMAN  
12400 WILSHIRE BOULEVARD  
SEVENTH FLOOR  
LOS ANGELES, CA 90025-1030

EXAMINER

VU, TUAN A

ART UNIT PAPER NUMBER

2193

DATE MAILED: 11/02/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No. 09/976,313	Applicant(s) HARRISON ET AL.	
	Examiner Tuan A. Vu	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 29 September 2005.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 2-6,8-12,14-18 and 21-25 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 2-6,8-12,14-18 and 21-25 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

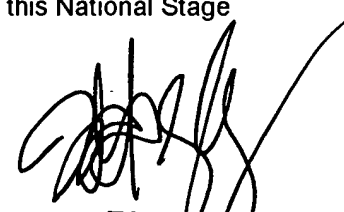
#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

  
TODD INGERBERG  
PRIMARY EXAMINER

#### Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.

- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_.

### DETAILED ACTION

1. This action is responsive to the Applicant's response filed 9/29/2005.

Claims 2, 8, 14, 21 have been amended. Claims 2-6, 8-12, 14-18, 21-25 are pending in the Office Action.

### *Claim Rejections - 35 USC § 103*

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 2-6, 8-12, 14-18 and 21-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cramer et al., USPN: 5,107,418 (hereinafter Cramer) in view of Archambault, USPN: 6,173,444 ( hereinafter Archambault).

**As per claim 2**, Cramer discloses a method comprising  
defining a definition-node and a use-node for a variable definition in intermediate language ( e.g. col. 3 lines 47-51; col. 4, lines 12-46; *program IR* - Fig. 1; col. 4, lines 14-16 –  
Note: create DUDe nodes reads on definition-node and use-node for each variable in a chain of the graph);

assigning an alias node for one or more aliases representing an equivalence class of memory accesses ( e.g. step 613, step 615 – Fig. 6 – Note: identification of one memory conflict operations as alias grouping aliased node together is equivalent to identifying one equivalence class – see *multiple\_alias* Appendix pg. 17-18);

Art Unit: 2193

introducing an edge into a dependence flow graph connecting a alias corresponding to a equivalence class associating a definition node (step 615, 617, 618: YES – Fig. 6); an

introducing an edge in a dependence flow graph connecting a alias corresponding to a equivalence class associating a use node (steps 616: YES, step 615 - Fig. 6); wherein the number of edges in the dependence flow graph is linear to a number of the nodes in the dependence flow graph ( Note \*: introducing a edge to a associating use node as repeated in the flow graph by Cramer reads on a number of edges therein being linearly dependent to a number of nodes therein)

But Cramer does not explicitly disclose assigning definition-node or use-node for one or more definition statements. But a node in a chain of statements in an intermediate language analysis in the art of analyzing data dependency and alias resolution entails defining statements the analysis of which leads to reference pointer issues that are to be addressed (see Fig. 2 and related text); hence the creation of def-use nodes for more definition statements is disclosed. Otherwise, such limitation would have been obvious because Cramer not only addresses pointer dereference in some equivalence class resolution but also mentions that the node to be created from the equivalence class resolution are involving various type of basic type ( col. 4, lines 30-37) and also works with a set of plural definition sets to address ( col. 6, line 67 to col. 7, line 6) and intends to address variable or dereferencing effects of called procedures (col. 2, line 66 to col. 3, line 2). Definition statements can be seen from block analysis ( see Cramer Fig. 2) and this is further disclosed in the intraprocedural or interprocedural by Archambault (e.g. col. 5, line 4-56) wherein analysis of statements defined for function calls therein involves r-val and l-val, i.e. one or more definition statements. In case the deferencing of variables in the deferencing

Art Unit: 2193

side effects (e.g. of procedure calls ) by Cramer is not considered a plurality of definition statements, it would have been obvious for one of ordinary skill in the art at the time the invention was made to expand the creation of def-node or use-node by Cramer so that it also applies to the inter/intra procedure creation of nodes in relation ( one or more definition statements) to alias setting when establishing dependency in control flow graph as taught by Archambault because set of objects that the compiler needs to keep track of amounts to more than single variables in order to resolve pointer dependency when resolving all aliases assumptions in increasingly complex program; and by approaching pointer issue by Archambault via set of related objects, the alias resolution in so doing can also reduce the size of program ( see Archambault, col. 2-3).

Nor does Cramer (combined with Archambault) explicitly state that the number of the edges in the dependence flow graph (see Note \*) is independent of a definition-use structure of the intermediate language program. As each of the definition or use node is created for a variable in the flow graph, Cramer teaches creating an edge connecting an alias corresponding to an equivalence class associating a definition node (step 615, 617, 618: YES – Fig. 6) and creating an edge in a dependence flow graph connecting an alias corresponding to an equivalence class associating a use node (steps 616: YES, step 615 - Fig. 6), hence creating edges only for those aliases corresponding to an equivalence class (Note: the number of aliases thus created is not directly proportional to the number of def-use structures, i.e. the number of edges introduced as in Note \* is independent from the number of definition/use structures being defined for variables in the flow graph from above -- as in Fig. 6). Hence, Cramer (in combination with Archambault) has disclosed the above limitation.

**As per claim 3**, Cramer discloses performing a memory alias analysis of said intermediate language program to partition the memory accesses of said intermediate language program into equivalence classes such that any two memory accesses that reference the same storage location belong to the same equivalence class ( e.g. *equivalenced or aggregate* - col. 5, lines 28-46 col. 6, lines 34-61; col. 7, line 48 to col. 8, line 9; step 114 –Fig 2).

**As per claim 4**, Pande discloses analysis using said dependence flow graph ( e.g. Fig. 1).

**As per claim 5**, Cramer discloses adding said alias-node to a set of nodes (Fig. 6 ) but does not disclose for each alias-node in the dependence flow graph assigning an initial value to the alias corresponding to said alias-node prior to the adding; but Archambault, discloses assigning a initial set of alias node with initial information saved therein (col. 5, lines 6-15 – step a); adding successors node to an alias-node (e.g. *pseudo* – col. 5, lines 18-23 – step c). Hence, this step of assigning an initial value to a alias would have been obvious for Cramer to use in order to keep track what actually has been used in the working set of alias nodes as approached by Archambault. In view of the teachings by Cramer (see step 613 –Fig 6) and Archambault to keep track with evolution of initial set of alias nodes so that data being resolved in the procedural call flow with actual updating of R-val upon resolution of alias set inside of procedures ( see Archambault, Fig. 1-3), the implementation by Archambault with updating of initial values being provided as an enhancement to Cramer's method for marching up and down basic blocks (see Appendix pg. 27-28) would have been obvious for one skill in the art at the time the invention was made because in doing so all the intraprocedural alias-related reference would be solved and propagate this into the interprocedural techniques as taught by Archambault to achieve the code

Art Unit: 2193

optimization of basic blocks and inter/intra procedural alias resolution techniques as intended by both Cramer and Archambault.

**As per claim 6**, Cramer does not disclose that said initial value comprises a set of abstract values to form a join-complete partial order. But official notice is taken that initializing each element in a set to be processed by an algorithm so that when the algorithm finished as complete the concept of joining the initial set with the recurring operations taking place during iterative steps of the algorithm was a known concept at the time the invention was made, and this is partially evidenced by Cramer (e.g. *removed ... from GEN set* - col. 6, lines 20-28; *sets, Aho, Sethi & Ullman* - col. 7, lines 32-41; Appendix B, pg. 9-10, *list in, out, NULL list*). And the iteration so to join the initial elements as results are obtained for updating the initial set is disclosed or strongly suggested by Archambault when creating alias node and associated R-val are updated ( see Fig. 1-3). Hence, by resolving the initial data being stored at the variables representing alias node, the concept of completing a partial order is disclosed, i.e. joining the initial value with the actual value and put forth the final result as the most significant of the two ( see Archambault, Fig. 2-3). Hence, this limitation would have been obvious by virtue of the rationale using the initial set as set forth in claim 5 above, combining Archambault's initial set updating to Cramer's basic block traversal and alias resolution.

**As per claims 8-10**, these claims correspond to claims 2-4, respectively; and are rejected with the corresponding rejection as set forth therein.

**As per claims 11-12**, these claims correspond to claims 5-6, respectively; and are rejected with the corresponding rejection as set forth therein.

Art Unit: 2193

**As per claims 14-16**, these claims are apparatus version of subject matter claimed in corresponding to claims 2-4, respectively; and are rejected with the corresponding rejection as set forth therein.

**As per claims 17-18**, these claims correspond to claims 5-6, respectively; and are rejected with the corresponding rejection as set forth therein.

**As per claim 21**, this claim is an apparatus claim incorporating means for performing the same steps as recited in claim 2 or 14; hence is rejected with the corresponding rejection as set forth therein.

**As per claim 22**, see claim 4.

**As per claim 23**, Cramer discloses adding said alias-node to a set of nodes (Fig. 6 ) but does not explicitly disclose having a initial set of nodes and remove a node from the set as each node is processed for adding in the dependence flow; but the concept to start a algorithmic process as mentioned by Cramer (e.g. *removed ... from GEN set* - col. 6, lines 20-28; sets , Aho, Sethi & Ullman - col. 7, lines 32-41; Appendix B, pg. 9-10, *list in, out , NULL list*) in order to go through every elements to be processed using a set and emptying it until all the elements therein have been processed was a known concept by algorithm execution engine or compilers. And this start set limitation for an algorithmic process until completion is disclosed; or else would have been obvious because based on Cramer's teachings from above one skill in the art would be motivated to use start set of elements containing some unresolved aliased node so that it would make the execution of the algorithm more visibly manageable thereby efficiently support the global variables dependency resolution in Cramer's def-use method until the loop is completed checked ( see Fig. 4-6).



Further, Cramer does not disclose a statement of the form PUT (A, E), even though Cramer discloses resolving via creating definition/use grouping in terms of Lval and set a flag ( Appendix: *bv-index, Lval* - pg. 13-14) from basic block defining of set in and out of the basic blocks; which is further enhanced by Archambault. That is, Archambault (e.g. col. 5, line 4-56) discloses analysis of statements defined for function calls therein involves r-val and l-val, i.e. one or more definition statements; hence disclose evaluating an expression to retrieve a r-val from a l-val, i.e. obtaining a evaluation of E to store it into A, or PUT into A, based on an initial set from which to update as Archambault progresses along in the reduction of the alias graph ( Fig. 2). It would have been obvious for one of ordinary skill in the art at the time the invention was made to enhance the r-val update as taught by Archambault to the alias tree generating by Cramer because by resolving the initial data being stored at the variables representing alias node in the grouping as mentioned by Cramer and reconciling the Lval( or A) with the correct Rval (or E) evaluation results ( see Archambault, Fig. 2-3), the tree traversal and dependency resolution of Cramer in terms of global variables resolving ( PUT based on E) or intra/interprocedural deference/reference issues can be improved as set forth in claim 6.

**As per claims 24-25**, these claims incorporate the rejection of claims 11 and 17 and further include the limitations that correspond to claim 23; and are rejected with the corresponding rejection as set forth therein.

#### ***Response to Arguments***

4. Applicant's arguments filed 9/29/2005 have been fully considered but the arguments are not convincing. Following are the Examiner's observations in regard thereto.

Art Unit: 2193

(A) Applicant has submitted that Cramer does not disclose or suggest ‘a number of the edges ... definition-use structure of the intermediate language program’ (Appl. Rmrks, pg. 9, 3<sup>rd</sup> para). The limitation as interpreted requires analysis and explanation in what is described in two parts as follows.

First, a non finite number of edges being created during the creation of the dependence flow graph for joining the equivalence class related alias nodes is linearly dependent of another non finite number of nodes of such graph. From this point of view, the nodes related to an alias class and for each of which a corresponding edge is to be connected to, as explained in the rejection, entail that there is direct proportionality between those number of edges and those above alias nodes. As set forth in the rejection, Cramer has been cited to show such linear dependency based on the creation of edges to link to each of the alias structures ( see Fig. 6 ).

Second, the number of edges thus created is independent from a non finite number of definition-use structures as these are conceived from the intermediate code. The rejection has shown that all definition or use nodes in Cramer’s method have been created for all variables of the intermediate code ( see col. 3 lines 47-51; col. 4, lines 12-46; *program IR* - Fig. 1; col. 4, lines 14-16); and since it is recognized from Fig. 6 that only those alias structures therein are connected to the number of edges (each alias node with one of such edge ) in the flow graph, there is no direct linear dependency or proportionality between the so-created number of edges ( joining the alias structure) and the systematic creation of variables-based definition or use node from above, i.e. the number of alias node being only but a subset of all the definition-use node or def-use structure which require a bigger number for most or all the variables of the IR code. Hence, the claim for being not so specific about how such dependency or independency is

Art Unit: 2193

implemented has allowed broad reasonable interpretation leading to the rationale as proffered above.

(B) Applicant has submitted that neither Cramer nor Archambault disclose or suggest the cited feature of claim 2 and similarly in claims 8, 14, and 21 (Appl. Rmrks, pg. 9, last para; pg. 10, top). The rejection has shown that the limitation at issue has been fulfilled as set forth above, in view of broad interpretation resulting from the lack of specificity in the claimed feature. The claims for lack of specificity in terms of defining the relationship encompassed by the terms used ( e.g. *dependent* or *independent* ) has apparently allow more than one way of construing the targeted subject matter and thus fail to preclude Cramer's ( in view of Archambault) teachings being cited from reading on the subject matter thus recited.

For the reasons as stated in the analysis provided in section A, the added limitation has been perceived as being met by the references used, the Applicant's arguments deemed non convincing in light thereof; and the claims stand rejected as set forth in the rejection.

### ***Conclusion***

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 ( for non-official correspondence – please consult Examiner before

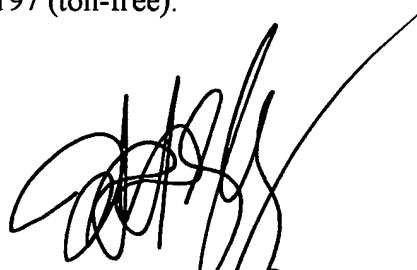
Art Unit: 2193

using) or 571-273-8300 ( for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT  
October 30, 2005



TODD INGBERG  
PRIMARY EXAMINER